TU Clausthal
Clausthal University of Technology

Model Checking Coalition Logic on Implicit Models is Δ_3^P-complete

Nils Bulling

IfI Technical Report Series IfI-10-02

Department of Informatics
Clausthal University of Technology

Impressum

Publisher: Institut für Informatik, Technische Universität Clausthal
Julius-Albert Str. 4, 38678 Clausthal-Zellerfeld, Germany
Editor of the series: Jürgen Dix
Technical editor: Michael Köster
Contact: michael.koester@tu-clausthal.de
URL: http://www.in.tu-clausthal.de/forschung/technical-reports/
ISSN: 1860-8477

The IfI Review Board

Model Checking Coalition Logic on Implicit Models is Δ_3^P-complete

Nils Bulling

Clausthal University of Technology, Department of Informatics, Germany
bulling@in.tu-clausthal.de

Abstract

In this note we show that model checking Coalition Logic over Concurrent Game Structures in which the transition function is given implicitly by a set of Boolean formulae is Δ_3^P-complete.

1 Introduction

Coalition Logic [Pauly, 2002] (**CL**) is a strategic logic that allows to model and to reason about one-step abilities of agents. It is well known, that **CL** can be understood as the next-time fragment of Alternating-time Temporal Logic (**ATL**) [Alur et al., 2002]. Hence, model checking **CL** is at most as hard as for **ATL**. From [Alur et al., 2002] we know that model checking **ATL** is P-complete over *Concurrent Game Structures* (CGS's). In [Bulling et al., 2010] it is shown that he proof of the lower bound can easily be modified to provide also a P-hardness proof for **CL**. All these results are with respect to the size of a given CGS which is defined as the number of transitions in the model.

Often, the number of transitions in a CGS is exponential in terms of states and agents. Taking this into account it is reasonable to encode the transition function symbolically resulting in a more compact model; that is, in a model of smaller size. We will call such models *implicit* CGS's. The size of such a model is measured with respect to the number of states and the size of the *encoded* transition function [Laroussinie et al., 2008]. Given this new representation/measure the model checking complexity of **ATL** is proven to be Δ_3^P-complete [Laroussinie et al., 2008, Jamroga and Dix, 2005] and [Jamroga and Dix, 2008]. In this note we prove that **CL** is also Δ_3^P-complete over implicit CGS's.

2 Coalition Logic: Syntax and Semantics

Firstly, we present the language $\mathcal{L}_{CL}$; subsequently, we introduce implicit CGS's and define a semantics for $\mathcal{L}_{CL}$. In the following let Π be a non-empty set of propositions and $\mathbb{A}\mathrm{gt} = \{1, \ldots, k\}$ be a non-empty and finite set of agents.

2.1 The language

Coalition Logic (**CL**), introduced in [Pauly, 2002], is a logic for modeling and reasoning about strategic abilities of agents. The main construct of **CL**, $[A]\varphi$, expresses that coalition A can bring about φ in a single-step game.

Definition 1 (Language $\mathcal{L}_{CL}$ [Pauly, 2002]) *The* language $\mathcal{L}_{CL}$ *is given by all formulae generated by the following grammar:* $\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid [A]\varphi$*, where* $\mathsf{p} \in \Pi$ *and* $A \subseteq \mathbb{A}\mathrm{gt}$*.*

In [Pauly, 2002], *coalitional models* were chosen as semantics for $\mathcal{L}_{CL}$. These models are given by (St, E, π) consisting of a set of states St, a *playable effectivity function* E, and a valuation function π. The effectivity function determines the outcome that a coalition is effective for, i.e., given a set $X \subseteq St$ of states a coalition C is said to be effective for X iff it can enforce the next state to be in X. However, in [Goranko and Jamroga, 2004] it was shown that *concurrent game structures* (CGS's) provide an equivalent semantics, and that **CL** can be seen as the *next-time fragment* of **ATL**.

2.2 Semantics

The semantics for $\mathcal{L}_{CL}$ is defined over a variant of transition systems where transitions are labeled with combinations of actions, one per agent. Formally, a *concurrent game structure* (CGS) is a tuple

$$\mathfrak{M} = (\mathbb{A}\mathrm{gt}, St, \Pi, \pi, Act, d, o)$$

which includes a non-empty finite set of all agents $\mathbb{A}\mathrm{gt} = \{1, \ldots, k\}$, a non-empty finite set of states St, a set of atomic propositions Π and their valuation $\pi : \Pi \to 2^{St}$, and a non-empty finite set of (atomic) actions Act. Function $d : \mathbb{A}\mathrm{gt} \times St \to 2^{Act}$ defines non-empty sets of actions available to agents at each state, and o is a (deterministic) transition function that assigns the outcome state $q' = o(q, \alpha_1, \ldots, \alpha_k)$ to state q and a tuple of actions $\langle \alpha_1, \ldots, \alpha_k \rangle$ for $\alpha_i \in d(i, q)$ and $1 \leq i \leq k$, that can be executed by $\mathbb{A}\mathrm{gt}$ in q. We also write $d_a(q)$ instead of $d(a, q)$. So, it is assumed that all the agents execute their actions synchronously: The combination of the actions, together with the current state, determines the next transition of the system.

A *strategy* of agent a is a conditional plan that specifies what a is going to do in each state; that is, a function $s_a : St \to Act$ where $s_a(q) \in d_a(q)$. The set of such strategies is denoted by Σ_a.

A *collective strategy* for a group of agents $A = \{a_1, \ldots, a_r\} \subseteq \mathbb{A}\mathrm{gt}$ is simply a tuple $s_A = \langle s_{a_1}, \ldots, s_{a_r} \rangle$ of strategies, one per agent from A. By $s_A|_a$, we denote agent a's part s_a of the collective strategy s_A where $a \in A$. The set of A's collective perfect information strategies is given by $\Sigma_A = \prod_{a \in A} \Sigma_a$.

Function $out(q, s_A)$ returns the set of all paths[1] that may occur when agents A execute strategy s_A from state q onward:

$out(q, s_A) = \{\lambda = q_0 q_1 q_2 \ldots \mid q_0 = q$ and for each $i = 1, 2, \ldots$ there exists a tuple of agents' decisions $\langle \alpha_{a_1}^{i-1}, \ldots, \alpha_{a_k}^{i-1} \rangle$ such that $\alpha_{a_j}^{i-1} \in d_{a_j}(q_{i-1})$ for every $a_j \in \mathbb{A}\mathrm{gt}$, and $\alpha_{a_j}^{i-1} = s_A|_{a_j}(q_{i-1})$ for every $a_j \in A$, and $o(q_{i-1}, \alpha_{a_1}^{i-1}, \ldots, \alpha_{a_k}^{i-1}) = q_i\}$.

The semantics for $\mathcal{L}_{CL}$ is shown below. Informally speaking, $\mathfrak{M}, q \models [A]\varphi$ if, and only if, there exists a collective strategy s_A such that φ holds in the next state on each computation from $out(q, s_A)$.

Definition 2 (Semantics $\models$) *Let $\mathfrak{M}$ be a* CGS. *The* semantics *for $\mathcal{L}_{CL}$, denoted by $\models$, is defined as follows:*

$\mathfrak{M}, q \models \mathsf{p}$ *iff* $\lambda[0] \in \pi(\mathsf{p})$ *and* $\mathsf{p} \in \Pi$*;*

$\mathfrak{M}, q \models \neg\varphi$ *iff* $\mathfrak{M}, q \not\models \varphi$*;*

$\mathfrak{M}, q \models \varphi \wedge \psi$ *iff* $\mathfrak{M}, q \models \varphi$ *and* $\mathfrak{M}, q \models \psi$*;*

$\mathfrak{M}, q \models [A]\varphi$ *iff there is a strategy $s_A \in \Sigma_A$ for A such that for every path $\lambda \in out(s_A, q)$, we have $\mathfrak{M}, \lambda[1] \models \varphi$.*

Formally, the logic **CL** is given by $(\mathcal{L}_{CL}, \models)$; that is, by the language $\mathcal{L}_{CL}$ and the semantics just introduced.

An *implicit concurrent game structure* (to the best of our knowledge, this has been introduced for the first time in [Laroussinie et al., 2008], but already present in the ISPL modeling language behind MCMAS [Raimondi and Lomuscio, 2004, Raimondi, 2006]) is defined similarly to a CGS but the transition function is encoded in a particular way often allowing for a more compact representation than the explicit transition table. Formally, an *implicit* CGS is given by $\mathfrak{M} = \langle \mathbb{A}\mathrm{gt}, St, \Pi, \pi, Act, d, \hat{o} \rangle$ where $\hat{o}$, the *encoded transition function*, is given by a sequence

$$((\varphi_0^r, q_0^r), \ldots, (\varphi_{t_r}^r, q_{t_r}^r))_{r=1,\ldots,|St|}$$

[1] A path λ is an infinite sequence of states such that subsequent states are connected by a transition. We use $\lambda[i]$ to refer to state q_i, i.e. $\lambda[i] = q_i$, provided that $\lambda = q_0 q_1 \cdots \in St^\omega$.

where $t_r \in \mathbb{N}_0$, $q_i^r \in St$ and each φ_i^r is a Boolean combination of propositions $\mathsf{exec}_\alpha^\mathsf{j}$ where $j \in \mathbb{A}\mathrm{gt}$, $\alpha \in Act$, $i = 1, \ldots, t$ and $r = 1, \ldots, |St|$. It is required that $\varphi_{t_r}^r = \top$. The term $\mathsf{exec}_\alpha^\mathsf{j}$ stands for "agent j executes action α". We use $\varphi[\alpha_1, \ldots, \alpha_k]$ to refer to the Boolean formula over $\{\top, \bot\}$ obtained by replacing $\mathsf{exec}_\alpha^{\mathsf{a_j}}$ with $\top$ (resp. $\bot$) if $\alpha_j = \alpha$ (resp. $\alpha_j \neq \alpha$).

The encoded transition function induces a standard transition function $o_{\hat{o}}$ as follows:

$$o_{\hat{o}}(q_i, \alpha_1, \ldots, \alpha_k) = q_j^i \text{ where } j = \min\{\kappa \mid \varphi_\kappa^i[\alpha_1, \ldots, \alpha_k] \equiv \top\}$$

That is, $o_{\hat{o}}(q_i, \alpha_1, \ldots, \alpha_k)$ returns the state belonging to the formula φ_κ^i (associated with state q_i) with the minimal index κ that evaluates to "true" given the actions $\alpha_1, \ldots, \alpha_k$. We also use $\hat{o}(q_i, \alpha_1, \ldots, \alpha_k)$ to refer to $o_{\hat{o}}(q_i, \alpha_1, \ldots, \alpha_k)$. Note that the function is well defined as the last formula in each sequence is equivalent to $\top$: no deadlock can occur. The size of $\hat{o}$ is defined as

$$|\hat{o}| = \sum_{r=1,\ldots,|St|} \sum_{j=1,\ldots,t_r} |\varphi_j^r|,$$

that is, the sum of the sizes of all formulae. Hence, the size of an implicit CGS is given by $|St| + |\mathbb{A}\mathrm{gt}| + |\hat{o}|$. Recall, that the size of an explicit CGS is $|St| + |\mathbb{A}\mathrm{gt}| + m$ where m is the number of transitions. Finally, we require that the encoding of the transition function is reasonably compact, that is, $|\hat{o}| \leq \mathbf{O}(|o_{\hat{o}}|)$.

3 Model Checking Complexity

Firstly, we recall two well-known results.

Theorem 1 ([Laroussinie et al., 2008, Jamroga and Dix, 2005]) *Model checking* **ATL** *over implicit* CGS*'s is Δ_3^P-complete with respect to the size of the model and the length of the formula.*

The Δ_3^P-hardness proof of [Laroussinie et al., 2008] uses the "nexttime" and "until" temporal operators in the construction of an **ATL** formula that is used in the reduction of SNSAT_2. We give a proof that uses only the language $\mathcal{L}_{CL}$.

Theorem 2 *Model checking* **CL** *over implicit* CGS*'s is Δ_3^P-complete with respect to the size of the model and the length of the formula.*

Proof. The upper bound follows from the result that model checking **ATL** is in Δ_3^P.

We extend the proof from [Laroussinie et al., 2008] such that only the next-time operator is used. The proof is done by reducing the Δ_3^P-complete problem SNSAT$_2$. A SNSAT$_2$ instance $\mathcal{I}$ consists of formulae

$$(\star) \qquad z_i = \exists X_i \forall Y_i \psi_i(z_1, \ldots, z_{i-1}, X_i, Y_i)$$

where $X_i = \{x_i^1, \ldots, x_i^s\}$ and $Y_i = \{y_i^1, \ldots, y_i^s\}$ are sets of variables and $s \in \mathbb{N}$ for $i = 1, \ldots, m$. Accordingly to the truth of the formulae ψ_i the value of each z_i is uniquely defined. A valuation of $\mathcal{I}$ is a mapping $v_{\mathcal{I}}$ assigning these unique values to each variable z_i. Moreover, if $v_{\mathcal{I}}(z_i) = \top$ we define $v_{\mathcal{I}}^{z_i} : X_i \rightarrow \{\top, \bot\}$ to be some valuation of the variables X_i that witnesses the truth of z_i. Note, that each z_i recursively depends on $z_{i-1}, \ldots, z_1$. In the following we will often omit the subscript $\mathcal{I}$.

We construct the following implicit CGS $\mathfrak{M}_{\mathcal{I}}$ for a given SNSAT$_2$ instance $\mathcal{I}$. Firstly, we introduce agents, each controlling one variable. There are agents a_i^j (one agent per variable x_i^j) with actions $\{\top, \bot\}$, b_i^j (one agent per variable y_i^j) with actions $\{\top, \bot\}$, c_i (one agent per z_i) with actions $\{\top, \bot\}$, and d (the "selector") with actions $\{1, \ldots, m\}$ for $i = 1, \ldots, m$ and $j = 1, \ldots, s$. We use A (resp. C and B) to denote the set of all agents a_i^j (resp. c_i and b_i^j).

The states of the model are given by states q_i and $\bar{q}_i$ (one per z_i) and the two states $q_\top, q_\bot$. States $\bar{q}_i$ are labelled with proposition neg and state $q_\top$ is labelled with sat.

Before giving the formal definition of the encoded transition function, we explain the role of the agents. Agents a_i^j (resp. b_i^j and c_i) determine the value of the variables x_i^j (resp. y_i^j and z_i). Action $\top$ (resp. $\bot$) sets them true (resp. false). Agent d has a more elaborated function. Once, all moves of the other agents are fixed, the agent can decide to "check" whether formula ψ_i holds regarding the actions of the other agents by executing action i. If the check is successful, the system goes to the winning state $q_\top$. If not, it goes to the losing state $q_\bot$. However, there are some exceptions to that which will be presented in the formal definition of the encoded transition function.

The part $(\varphi_0^i, q_0^i), \ldots, (\varphi_{t_i}^i, q_{t_i}^i)$ in the encoded transition function associated with state q_i is defined as follows (where ψ_i' denotes the formula ψ_i in $(\star)$ in which each occurrence of x_i^j (resp. y_i^j and z_i) is replaced by $\mathsf{exec}_\top^{\mathsf{a_i^j}}$ (resp. $\mathsf{exec}_\top^{\mathsf{b_i^j}}$ and $\mathsf{exec}_\top^{\mathsf{c_i}}$) (recall, that $\mathsf{exec}_\alpha^{\mathsf{a}}$ means that agent a executes action α)):

$$(\mathsf{exec}_\mathsf{k}^\mathsf{d} \wedge (\wedge_{j=i-1,\ldots,k} \mathsf{exec}_\top^{\mathsf{c_j}}) \wedge \psi_k', q_\top)_{k=i,\ldots,1}, \tag{1}$$

$$(\mathsf{exec}_\mathsf{k}^\mathsf{d} \wedge (\wedge_{j=i-1,\ldots,k} \mathsf{exec}_\top^{\mathsf{c_j}}), q_\bot)_{k=i,\ldots,1}, \tag{2}$$

$$(\mathsf{exec}_\mathsf{k}^\mathsf{d} \wedge \neg \mathsf{exec}_\top^{\mathsf{c_k}}, \bar{q}_k)_{k=i-1,\ldots,1}, \tag{3}$$

$$(\top, q_\top) \tag{4}$$

Moreover, there are loops at states $q_\top$ and $q_\bot$ and transitions from $\bar{q}_i$ to q_i for $i = 1, \ldots, m$. The following lemma is fundamental to our reduction.

Lemma 3 *Let* $\chi_0 = \top$ *and*

$$\chi_{r+1} = [A \cup C](\mathsf{sat} \vee (\mathsf{neg} \wedge [\emptyset]\neg\chi_r))$$

for $r = 0, \ldots, m-1$ *where* sat *and* neg *are propositional symbols. Then, for all* $i \leq m$ *and* $r \geq i$ *it holds that*

$$\mathfrak{M}_{\mathcal{I}}, q_i \models \chi_r \text{ iff } v_{\mathcal{I}}(z_i) = \top.$$

Proof of Lemma. We proceed by induction on i. Firstly, we consider the base case $i = 1$.

"$\Rightarrow$": Suppose that $\mathfrak{M}, q_1 \models \chi_r$ for $r \geq 1$. Due to the definition of the transition function only rules (1,2,4) are present; hence, only $q_\top$ and $q_\perp$ are reachable. That is, the formula $\mathfrak{M}, q_1 \models [A \cup C]\mathsf{sat}$ must be satisfied (as the label neg cannot become true). But then, there must be a valuation of the x_1^j's such that for all valuations of the y_1^j's, ψ_1 evaluates true; hence, $v(z_1) = \top$.

"$\Leftarrow$": Suppose $v(z_1) = \top$. Then, there is a valuation of the variables x_1^j such that for all valuations of y_1^j the formula ψ_1 evaluates true. It is easily seen that the strategy in which each agent in A plays according to the valuation given by v^{z_1} and c_1 plays $\top$ witnesses that $q_1 \models [A \cup C]\bigcirc\mathsf{sat}$ (and thus also $\mathfrak{M}, q_1 \models \chi_r$ for $r \geq 1$).

For the inductive step suppose the assumption holds up to index $i \geq 1$.

"$\Rightarrow$": Suppose $\mathfrak{M}, q_{i+1} \models \chi_{r+1}$ for $r \geq i$. Firstly, we prove the following claim.

Claim: *Suppose* $\mathfrak{M}, q_{i+1} \models \chi_{r+1}$*, then each* c_l *with* $l \leq i$ *plays according to the valuation* $v(z_l)$.

Proof of claim. Suppose c_l plays $\perp$ and d plays l. Then, the next state of the system is $\bar{q}_l$ and consequently, $\mathfrak{M}, q_l \models \neg\chi_r$ and by induction hypothesis $v(z_l) = \perp$.

The other case is proven by induction. Suppose $i = 1$, $\mathfrak{M}, q_2 \models \chi_{r+1}$, and c_1 plays $\top$. We have to show that $v(z_1) = \top$. Suppose the contrary. Then, for any strategy of $A \cup C$ there is a strategy of B such that ψ_1' evaluates false. Hence, if d plays 1 rule (2) is firing and the next state is $q_\perp$ and thus $\mathfrak{M}, q_2 \not\models \chi_{r+1}$. Contradiction!

For the induction step, suppose that all agents c_l for $l < i$ play according to $v(z_l)$, that $\mathfrak{M}, q_{i+1} \models \chi_{r+1}$, and $c_i = \top$. We show that $v(z_i) = \top$. For the sake of contradiction, suppose that $v(z_i) = \perp$. Again, for any strategy of $A \cup C$ witnessing χ_{r+1} we have that there is a strategy of B that falsifies ψ_i' (note, that by assumption $c_1, \ldots, c_{i-1}$ play according to $v(z_1), \ldots, v(z_{i-1})$). So, if d plays i rule (2) is firing and the next state is $q_\perp$ which implies $\mathfrak{M}, q_{i+1} \not\models \chi_{r+1}$. Contradiction! ■

Now let s_{AC} be the strategy of agents $A \cup C$ that witnesses χ_{r+1} in q_{i+1}. Suppose player d plays $i+1$. Irrelevant of the move of c_{i+1} either rule (1) or rule (2) is firing. This does only depend on the valuation of ψ'_{i+1}. By assumption, we must have that ψ'_{i+1} is true for all strategies of B else $\mathfrak{M}, q_{i+1} \not\models \chi_{r+1}$. Because of the previous claim, we must also have that $v(z_{i+1}) = \top$.

"$\Leftarrow$": Suppose $v(z_{i+1}) = \top$. Let s_{AC} be the strategy in which players c_j play according to $v(z_j)$ and players a_j^o play according to v^{z_j} if $v(z_j) = \top$ and arbitrarily if $v(z_j) = \bot$ for $o = 1, \ldots, s$. Suppose player d plays $l \leq i+1$. Now, if each c_j for $j = i, \ldots, l$ plays $\top$ we have that ψ'_l is true as there is no valuation of variables Y_l that makes ψ_l false given the choices of $A \cup C$; hence, the next state is $q_\top$. Secondly, if d plays l and there is some agent c_j, $j > l$, that plays $\bot$; then rule (4) fires and the next state is also $q_\top$; the same holds if d plays $l > i+1$. Finally, suppose d plays l and $c_l = \bot$. Then, by the definition of the actions of agents C, $v(z_l) = \bot$ and by induction hypothesis $\mathfrak{M}, q_l \models \neg\chi_r$; thus, $\mathfrak{M}, \bar{q}_l \models \mathsf{neg} \wedge [\emptyset]\neg\chi_r$ is true. Taking all theses cases together we have $\mathfrak{M}, q_{i+1} \models \chi_{r+1}$. ■

This gives us the following polynomial reduction:

$$z_m = \top \text{ iff } \mathfrak{M}_{\mathcal{I}}, q_m \models \chi_m$$

■

4 Conclusions

We have shown (Theorem 2) that model checking $\mathcal{L}_{CL}$ over *implicit* CGS's is already Δ_3^P-complete; thus, resides in the same complexity class as model checking the more expressive language $\mathcal{L}_{ATL}$. This mirrors the situation for (explicit) models over which model checking each of these two logics is P-complete.

References

[Alur et al., 2002] Alur, R., Henzinger, T. A., and Kupferman, O. (2002). Alternating-time Temporal Logic. *Journal of the ACM*, 49:672–713.

[Bulling et al., 2010] Bulling, N., Dix, J., and Jamroga, W. (2010). Model checking logics of strategic ability: Complexity. In Dastani, M., Hindriks, K. V., and Meyer, J.-J. C., editors, *Specification and Verification of Multi-Agent Systems/Programs*. Springer.

[Goranko and Jamroga, 2004] Goranko, V. and Jamroga, W. (2004). Comparing semantics of logics for multi-agent systems. *Synthese*, 139(2):241–280.

[Jamroga and Dix, 2005] Jamroga, W. and Dix, J. (2005). Do agents make model checking explode (computationally)? In Pěchouček, M., Petta, P., and Varga, L., editors, *Proceedings of CEEMAS 2005*, volume 3690 of *Lecture Notes in Computer Science*, pages 398–407. Springer Verlag.

[Jamroga and Dix, 2008] Jamroga, W. and Dix, J. (2008). Model checking abilities of agents: A closer look. *Theory of Computing Systems*, 42(3):366–410.

[Laroussinie et al., 2008] Laroussinie, F., Markey, N., and Oreiby, G. (2008). On the expressiveness and complexity of atl. *CoRR*, abs/0804.2435.

[Pauly, 2002] Pauly, M. (2002). A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166.

[Raimondi, 2006] Raimondi, F. (2006). *Model Checking Multi-Agent Systems*. PhD thesis, University College London.

[Raimondi and Lomuscio, 2004] Raimondi, F. and Lomuscio, A. (2004). Automatic verification of deontic interpreted systems by model checking via OBDD's. In de Mántaras, R. and Saitta, L., editors, *Proceedings of ECAI*, pages 53–57.